

AKIŞ KONTROL MEKANİZMALARI

▼ Nedir?

- Kodun akışında belirli şartlara göre farklı yönlendirmeleri yapmamızı ve farklı algoritmaları yapılanmaları çalıştırmaızı sağlayan mekanizmalardır.
 - if-else
 - switch yapısı
- Bu iki yapı, aynı işi iki farklı şekilde yapmamızı sağlar. Aralarında teknik olarak fark olsa da işleyiş açısından fark yoktur.

▼ Swith case

▼ Switch case nedir?

- Kodun akışında belirli bir şarta göre yönlendirme yapmamızı sağlayan yapılanmadır.
- Bir değişkenin değerini **eşitlik durumlarını kontrol** ederken kullanılabilir.
- Prototip:

```
switch ( ) //eşitlik durumu kontrol edilecek olan deger bu  
{ //Kontrol edilen degerin turu ne ise case bloklarında d  
// Buradaki deęer statik ya da deęişken olabilirken, case:  
    case value1:  
        // switch durumunda verilen deęerin aşıtlıj durumunla  
        break;  
  
    case value2: // case value belirtildikten sonra : kalı  
  
        break; //case ile break arasına yazılan kodlar ilgili  
  
default:
```

```
// hiçbir case break arasındaki şart sağlanmıyorsa, default  
// default kalıbı zorunlu değildir.
```

```
break:  
}
```

▼ when keyword'ü

Switch yapılanmasında sadece elimizdeki değerın eşitlik durumunu kontrol edebilmekteyiz. Bunun dışında farklı şartları değerlendirmek istiyorsak when kullanırız.

```
switch (satisTutarı)  
{  
    case 100 when ..... : // satıl tutarı 100 ve whendei  
}
```

▼ goto keyword'ü

switch case de sadece eşitlik durumu incelediğimizden mantıksal işlem gerçekleştirmiyoruz. Dolayısıyla bazen farklı değerlere eşit olma durumunda aynı operasyonu kullandığımız seneryo ile karşılaşabiliirz

Farklı eşitliklerde kod tekrarına girmemek için goto keywordü ile git şu case i çalıştır diyebiliriz. Kodlar arası zıplama sağlar.

```
switch (  
{  
    case 5:  
        cw(i*5)  
        break;  
    case 6:  
        cw(i/2)  
        break;  
    case 10:  
        cw(i*5)// durumunda goto kullanabiliriz.
```

```

        //goto case 5; şeklinde. Bu durumda break komutu l
//birden fazla aynı case e yönlendirdiğimizde tek şekilde
        case 7:
        case 11:
            goto case 5; //şeklinde.

    }

```

▼ switch expressions

▼ if yapılanması

▼ if yapısı

- Şarta göre bir kodun çalışıp çalışmadığını belirleyen yapıdır.
- Elimizdeki bir değerin eşitlik durumu da dahil aklımıza gelebilecek tüm kıyas ve karşılaştırmaları yapabilen bir kontrol mekanizmasıdır.
- Tek satırlık bir işlem yapılacaksa scope kullanmadan da yazılabilir.
-

```

if() // her daim boolean türde olmalıdır. Mantıksal ve ka
{

}

// else kullanmak zorunda değiliz. Eğer if koşulunu sağlar

```

- Bool türde bir karşılaştırma yaparken `x= true;` `if(x== true)` ise şeklinde yazmak zorunda değiliz. `if(x)` yazmamız yeterli olur.

▼ if else yapısı

- Şarta göre bir kodun çalışıp, şartın olmadığı durumda bir başka kodun çalışmasını bekleyen bir yapılanmadır.

- Şartın değil durumunda da çalışacak kodu belirlemiş oluruz.

```

•   if(şart)
      {
          //şart true ise burası
      }

      else
      {
          // false ise burası tetiklenir.
      }

```

-

▼ if - else if

- Birden fazla şartı kontrol etmemizi sağlayan yapılanmadır.
- Eğer hava yağmurluysa şemsiye al, değilse alma. if else yapısı
- Eğer hava yağmurluysa şemsiye al, kapalıysa mont giy, yapmurlu değilse şemsiye alma..... if else if yaapısı

-

```

if(şart)
{
    //doğruysa burası tetiklenir.
}

else if(şart)
{
    //yukarıdaki şart doğrulanmadıysa ve bu doğruysa burası t
}

```

```
else if(şart)
{
//yukarıdaki şartlar doğrulanmadıysa ve bu doğruysa burası:
}

// yalnızca biri doğrulanabilir.
```

- Eğer mantıksal olarak iki şartı da sağlasın istiyorsam örneğin kullanıcıdan alınan bir sayı 20-30 arasındaysa A yaz, 20-40 arasındaysa B yazsın istiyorsam bunu if- else if ile yapamam. Çünkü A yazar, diğer şarta bile bakmadan blok içinden çıkar. O yüzden if (), if() blokları içerisinde kontrol etmeliyim ki tüm şartları tek tek gezsin ve A B çıktısını verebilsin.

-
-

▼ if - else if -.....- else

Birden fazla şartı kontrol edip , eğer hiç biri değilse else i çalıştırır.